A Mechanism for Creating Scientific Application Services on Demand from Workflows

Gopi Kandaswamy

Dennis Gannon

(Indiana University)

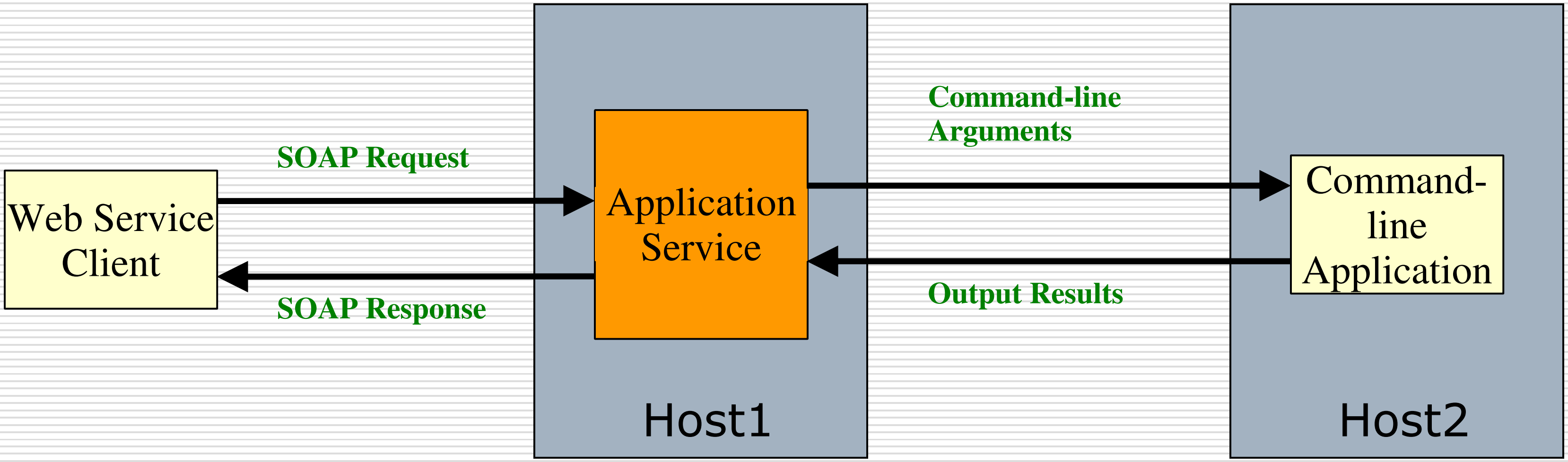
Outline

- □ Introduction
- □ Motivation
- □ Solution
- □ Performance & Scalability
- □ Applications

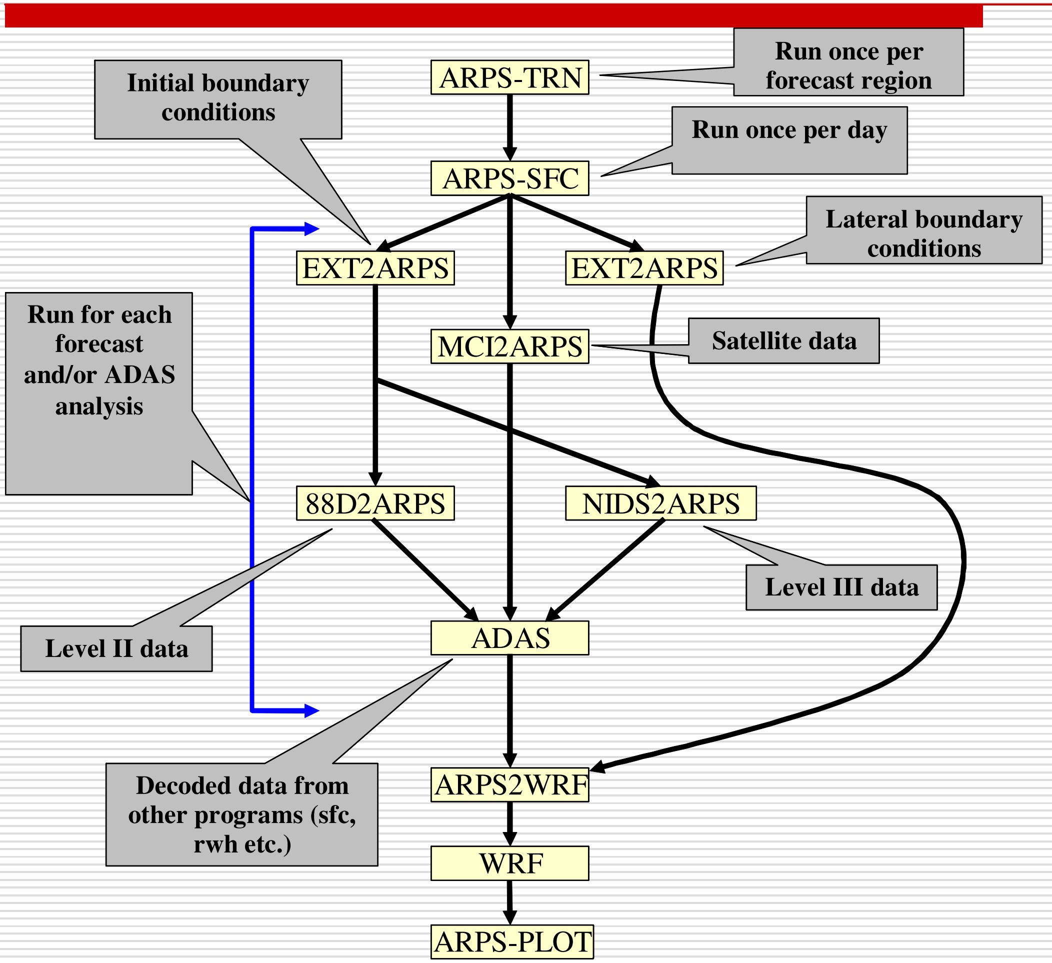
Web Services in Scientific Communities

- Web services are used to “wrap” scientific applications to
  - Describe, publish, discover and consume scientific applications in a standard way
  - Compose complex workflows from scientific applications
  - Run and monitor complex workflows on distributed resources
- Such web services that “wrap” scientific applications are called “application services”

An Application Service



A Scientific Workflow using Application Services


Initial boundary conditions
ARPS-TRN
Run once per forecast region
ARPS-SFC
Run once per day
EXT2ARPS
EXT2ARPS
Lateral boundary conditions
Run for each forecast and/or ADAS analysis
MCI2ARPS
Satellite data
88D2ARPS
NIDS2ARPS
Level III data
Level II data
ADAS
Decoded data from other programs (sfc, rwh etc.)
ARPS2WRF
WRF
ARPS-PLOT

The Problem

- Application services may not be available during a workflow execution
  - Unreliable resources (software, computers, networks)
  - Heavy load on service
  - Does not meet QoS or security requirements of client
- Workflows cannot complete unless all services are available

Current Solutions

- Provide backup services
  - Results in too many running services and wastes resources
- Make all services persistent
  - Requires a lot of resources and support infrastructure
  - Too expensive

Our Solution

- A Generic Application Factory
  - A persistent web service that knows how to create instances of any application service
- Use a Generic Application Factory to create instances of application services on-demand from workflows

Challenges

How do we create application services on-demand from workflows?

- From where do we get the application service binary?
- From where do we get a web service container to host the application services?
- How to reduce the overhead of creating application services on-demand?

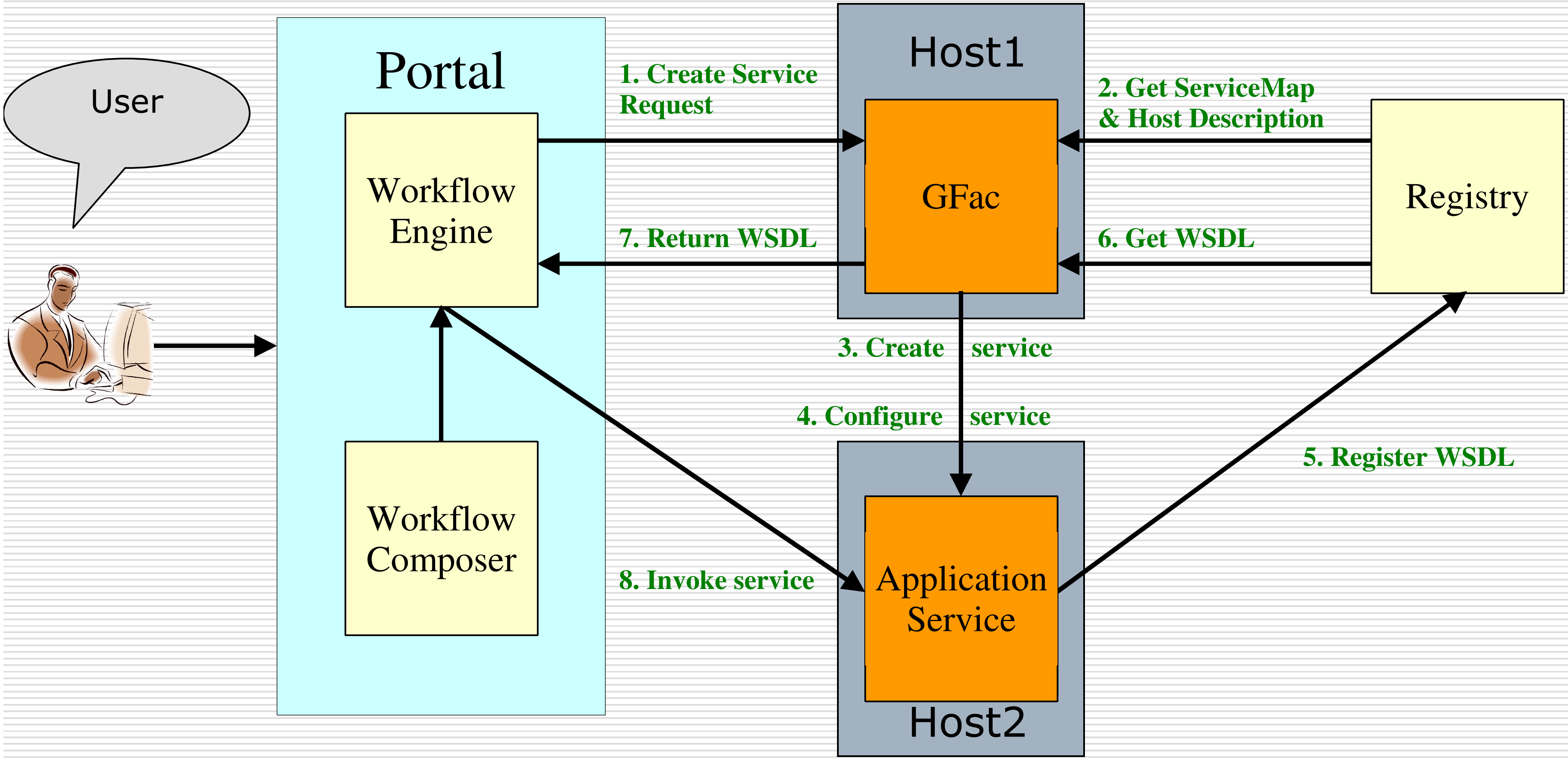
Our Implementation

- □ The Generic Application Factory (GFac)
- □ The Generic Service Toolkit: A toolkit that can “wrap” any command-line application as an application service
  - ■ Without writing any web service code
  - ■ Without modifying the application in any significant way

Creating an Application Service on Demand (1/2)

- Write “ServiceMap” document to describe your service
- Write “Application Deployment Description” document to describe a deployment of your application
- Upload the above two documents to a Registry service

Creating an Application Service on Demand (2/2)

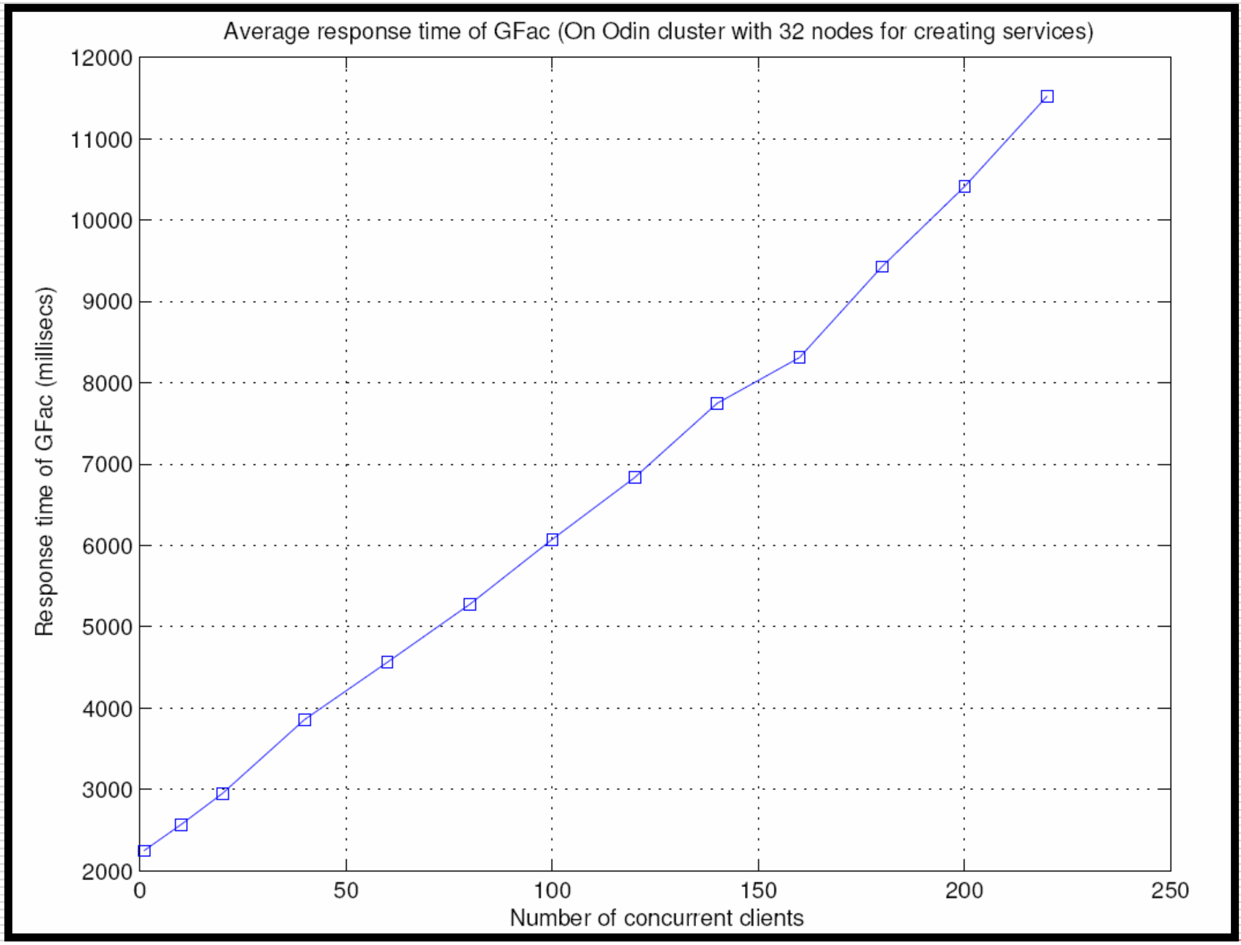


Performance of GFac

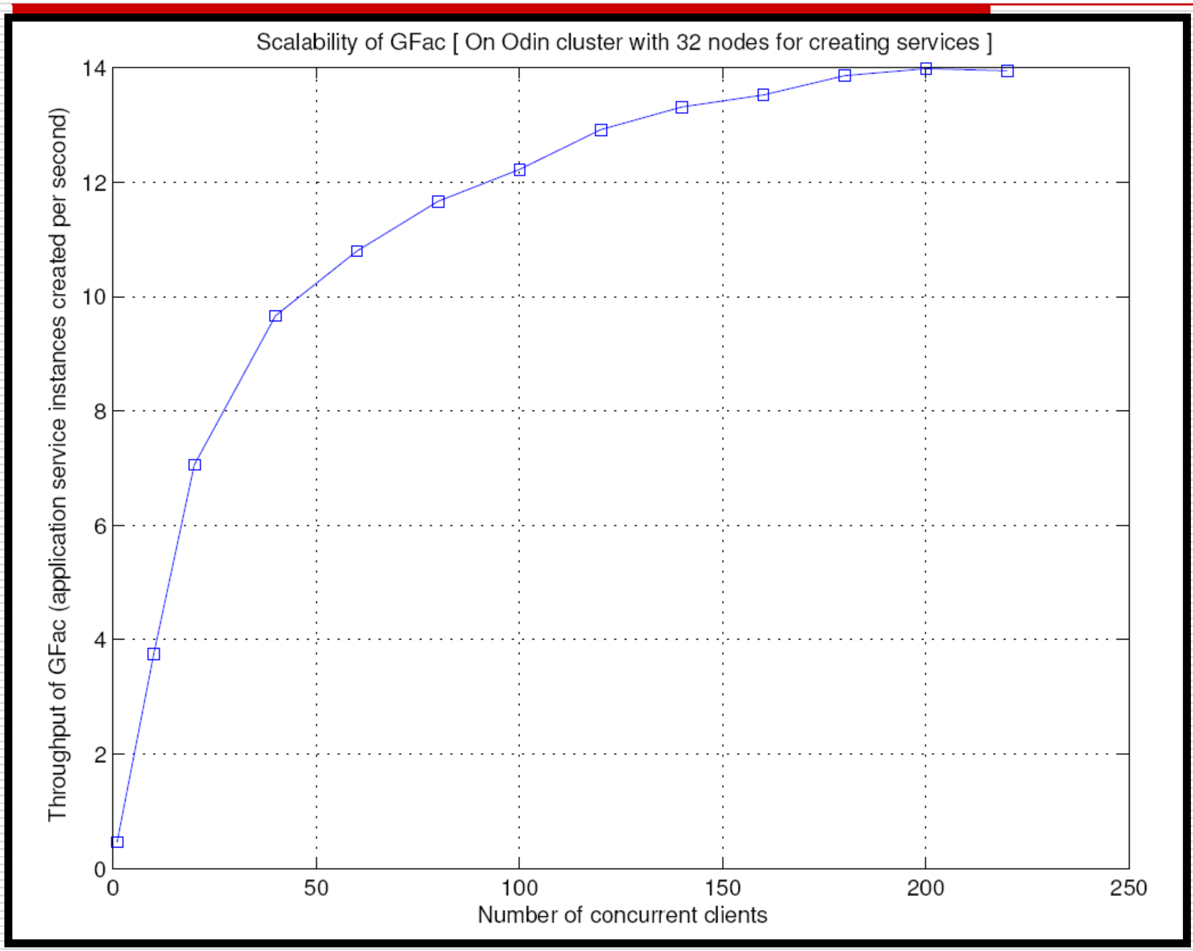
- GFac running on a node
- Web service client on another node
- Client sends message to GFac to create one application service
- Time elapsed between sending request and receiving response is measured

Security Mode	*Time (seconds)*
Insecure	2.2
TLS	2.5
MSG-SIG	3.0

Response Time of GFac (Insecure)

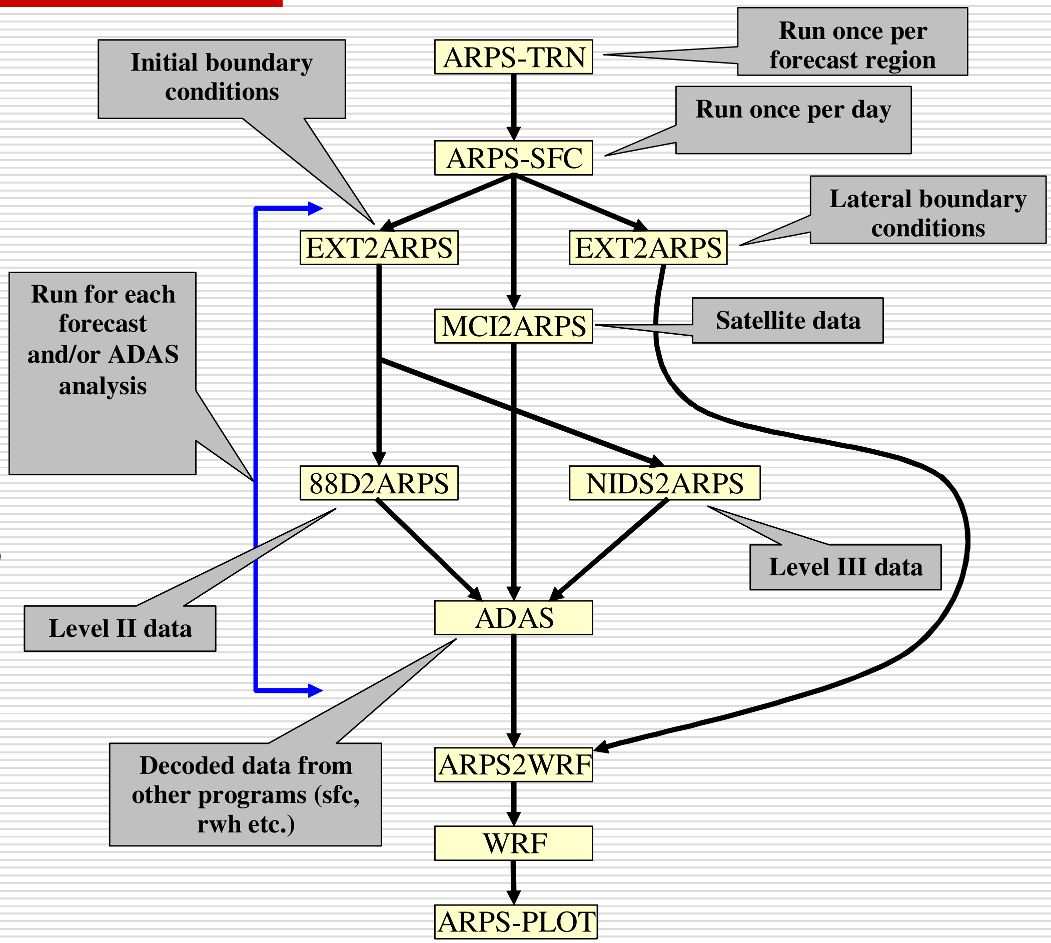


Scalability of GFac (Insecure)



LEAD: Linked Environments for Atmospheric Discovery

- □ Dynamically adaptive weather forecasting
- □ Through dynamic workflow orchestration
- □ Using web services



ARPS-TRN
Run once per forecast region
ARPS-SFC
Run once per day
Initial boundary conditions
EXT2ARPS
EXT2ARPS
Lateral boundary conditions
Run for each forecast and/or ADAS analysis
MCI2ARPS
Satellite data
88D2ARPS
NIDS2ARPS
Level III data
Level II data
ADAS
Decoded data from other programs (sfc, rwh etc.)
ARPS2WRF
WRF
ARPS-PLOT

ADaM: Algorithm Development and Mining System

- 100+ components that can be configured to do customized data mining tasks
- Several components like Feature Extraction, Data Transformation, Data Normalization and Classification were wrapped as web services and used in several small data mining workflows

RENCI Science Gateway

- □ To use the Generic Service Toolkit to make available 140+ bio-informatics applications as application services
- □ Enable integration with NIH, NSF and North Carolina funded services into workflows from the RENCI Science Gateway

Summary

- GFac and Generic Service Toolkit can be used
  - To “wrap” any command-line application as a secure and scalable application service
  - To create any application service on-demand
- Reduces cost of providing “highly available” application services

Questions